Relationship between the constant L and convergence rate, comparison w/ [23] (Reviewers $\#1$ and $\#3$). Th 3.3 (line 202) relates the algorithm convergence rate to the stepsize t_k. This stepsize depends on the constant L through the expression of $\mathrm{step}_{\#}(p)$, which shows the dependency between convergence rate and L. In the paper, we prove that $\mathrm{step}_{\#}(p)$ is lower bounded. We have observed numerically that the stepsize is larger than the stepsizes used by other discretizations schemes in the heavy-ball method, as shown in Fig 3, and will provide further numerical evidence in the revised version. We are currently working on an analytical comparison with [23], which requires the explicit computation of a tight lower bound of $\mathrm{step}_{\#}(p)$ as a nonlinear function of s, α, μ, and L (R$\#3$).

Omitted relevant literature (Reviewer $\#2$). We respectfully disagree with this comment, as we include the work by the authors suggested by R$\#2$, see [3], [22], [23], [28] and [29]. It is impossible to provide an exhaustive literature review, but we will include recent papers[1] (available after the submission of our work) by Attouch, França, and co-authors which deal, resp., with the discretization of inertial systems with Hessian-driven damping and conformal Hamiltonian systems to obtain optimization algorithms. We will include Kolarijani *et al.*, which uses hybrid dynamical systems to generate fast optimization methods that employ constant-stepsize discrete dynamics. The differences with our work are clear, as none of these references design variable-stepsize integrators based on event-triggered control.

Perceived limited applicability of the proposed setting and extensions beyond it (all reviewers). As a result of the concerns raised by R$\#2$, we have realized that the twice differentiablity assumption can be weakened: in the heavy-ball case, only continuous differentiablity is needed for the discretization. In Nesterov's case, twice differentiability arises from the presence of a Hessian term $\sqrt{s}\nabla^2 f(x)v$ in the ODE, which is inherited by the discretization. The work [23] replaces it by $\nabla f(x_{k+1}) - \nabla f(x_k)$ when discretized, providing an appealing research direction circumventing the use of the Hessian. It is standard practice in the literature to assume knowledge of μ and L for strongly-convex functions when looking for the optimal rate. Besides, several methods have been designed to approximate these constants in practice, and they can surely be adapted to our setting. As pointed by R$\#1$, the function g is case-dependent, but the methodology presented here is applicable to the discretization of other dynamical systems endowed with a Lyapunov function certificate. We agree with R$\#3$ that pursuing this will broaden the applicability of our theory. Although regularization can also be used to endow convex functions with strong convexity, it would also be extremely interesting to extend this methodology to the convex framework (R$\#2$). Nonetheless, the main point of the paper is to introduce the idea of a systematic way to develop discretizations that maintain the convergence rate properties of their continuous counterparts. R$\#1$ and $\#3$ point out that the originality of the paper is " basically beyond doubt" and we believe it may inspire new research given the recent explosion of activity in the area of high-resolution ODEs.

Importance of opportunistic state-triggered control and variable-stepsize discretization (Reviewer $\#3$). Opportunistic state-triggered control saves resources by taking into account the current system state while maintaining performance guarantees. This is in contrast to periodic sampling, where worst-case scenarios have to be taken into account, drastically reducing inter-sampling time. Analogously, the proposed integrators take into account the current state of the dynamics through the values of v and ∇f to adjust its stepsize while satisfying convergence and performance guarantees. This contrasts with fixed-stepsize integrators, whose stepsize is limited by the most unfavorable situation. In practice, this may have a critical impact on performance.We will address any possible confusion (especially regarding terminology) pointed by R$\#3$ in the revised version.

Simulations (all reviewers). We will include richer numerical experiments if the paper is accepted. We have run now simulations with quadratic functions defined by 50x50 matrices with similar results. *Convergence* will be shown by plotting the decay of the objective and Lyapunov functions (R$\#1$). Regarding *Fig* 2, we show that the three discretization procedures follow the same trajectory (the continuous dynamics). The proposed approach is able to follow the curve taking longer stepsizes, thus making further progress when run for an equal number of iterations. Formally, let us denote by t_k the stepsize of our method at iteration k and by s the stepsize of a fixed-stepsize integrator. After n iterations, our integrator approximates the continuous dynamics at $\sum_{k=1}^{n} t_k$, while the constant-stepsize integrators approximates it at $n \cdot s$. In simulations, $\sum_{k=1}^{n} t_k$ is significantly larger than $n \cdot s$ (R$\#1$). We will also include the ET integrator for comparison in Fig 2 in the revised version (R$\#3$). Finally, we introduce the *optimal stepsize* only for comparison purposes, as the minimizer is in practice unknown. Knowledge of the minimizer x_* would enable the explicit computation of the Lyapunov function (cf. Th. 3.1), which in turn allows to solve $\dot{V} + \alpha V = 0$ (cf. line 181) by any standard numerical method at any iteration. We refer to this solution as optimal stepsize (Fig 3, green), as is the actual largest stepsize one may take conserving the Lyapunov decay. Fig 3 illustrates how our algorithm is able to chase this optimal stepsize at any iteration, without knowledge of the minimizer (R$\#1$ and $\#3$). R$\#2$ also points out that the computation of the stepsize may be convoluted. While the ET integrator is more involved, the ST integrator relies on a simple function of the quantities $\|v\|$, $\|\nabla f\|$ and $\langle v, \nabla f\rangle$, (see step_{ST}, line 196) which can be computed easily.

[1] H. Attouch, Z. Chbani, J. Fadili, and H. Riahi. First-order optimization algorithms via inertial systems with Hessian driven damping. *arXiv:1907.10536*, 2019; G. França, J. Sulam, D. Robinson, and R. Vidal. Conformal symplectic and relativistic optimization. *arXiv:1903.04100*, 2019; A. S. Kolarijani, P. M. Esfahani, and T. Keviczky. Fast Gradient-Based Methods with Exponential Rate: A Hybrid Control Framework. In *Proceedings of the 35th International Conference on Machine Learning*, 2018